Towards Optimal Capacity Segmentation with Hybrid Cloud Pricing

Wei Wang, Baochun Li, Ben Liang
Department of Electrical and Computer Engineering
University of Toronto

IaaS clouds offer multiple pricing options

On-demand (pay-as-you-go)

- Static hourly rate x run hours $= p_r t$

Subscription (reservation)

- One-time subscription fee
- Free/discounted usage fee during the reservation period

Auction-like pricing (spot market)

- Users bid for computing instances
- No service guarantee

IaaS clouds offer multiple pricing options

GoGrid,
ElasticHosts,
BitRefinery,
Ninefold
...

- On-demand
- Subscription

Amazon EC2

- On-demand
- Subscription
- Auction-like pricing

Why multiple pricing?

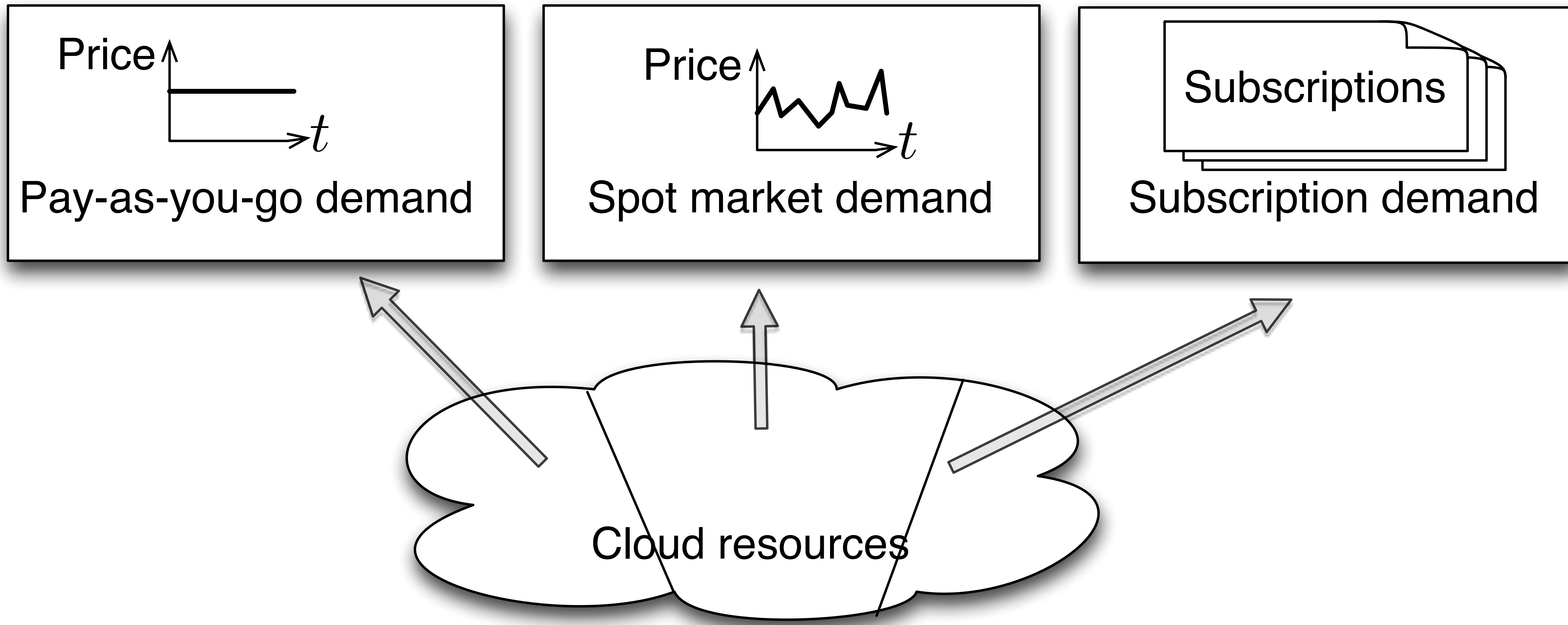
Compensate the deficiency of individual pricing

- Static pricing: awkward to market dynamics, easy to understand, risk-free with a static price
- Spot price: agile to demand/supply changes, hard to understand, risky due to price fluctuations

Expand the market demand

- Long-term users go for subscription
- Price-sensitive users bid in the spot market

How do cloud providers allocate its capacity to different pricing channels?

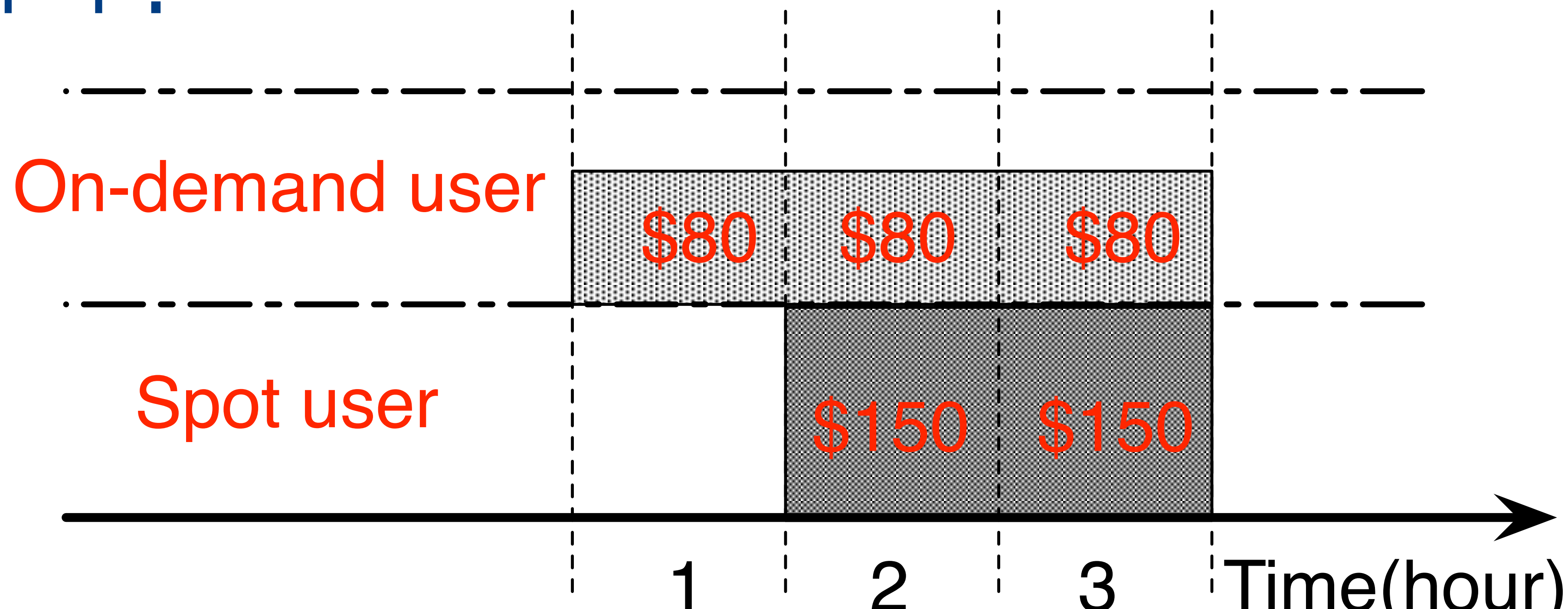


How to set the prices?

How many instances to offer in each pricing channel?

Objective: Revenue maximization

How many instances to offer in each channel in hour 1?

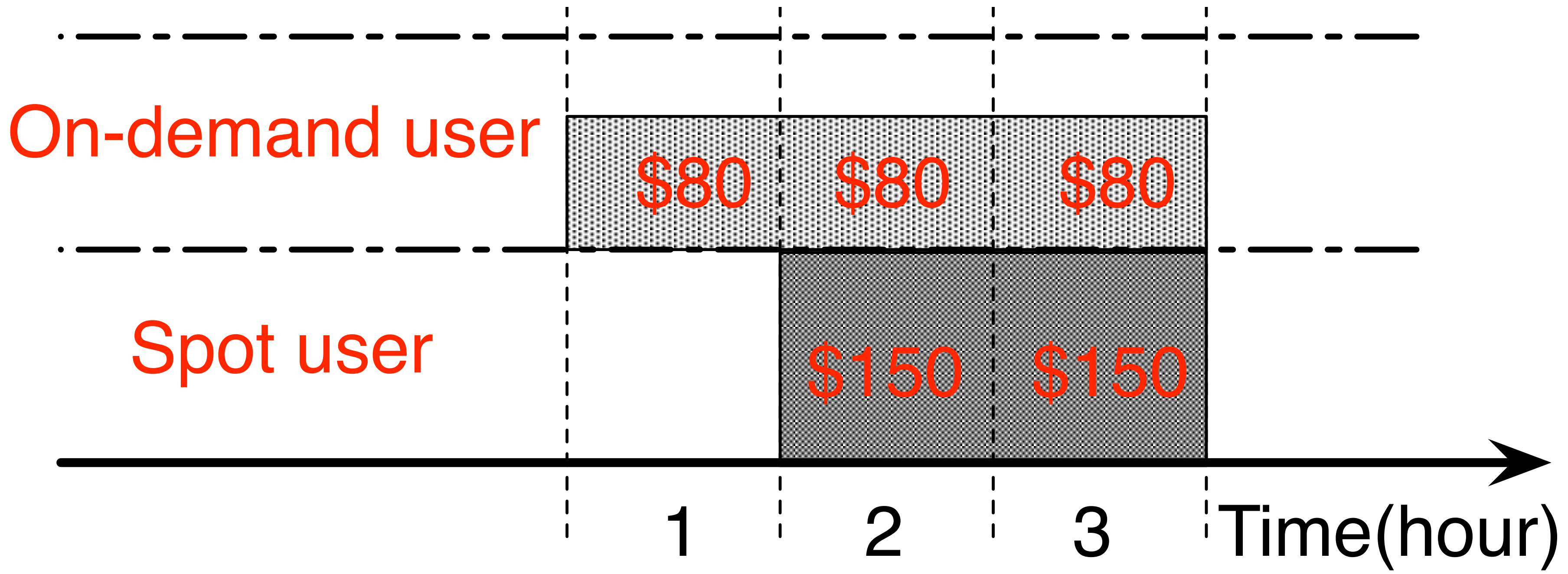


An on-demand user requests 80 instances for 3 hours, starting from hour 1, with on-demand rate $1

A spot user bids for 100 instances each at $1.5 per instance-hour, starting from hour 2

The available capacity of a cloud can only support 100 additional instances

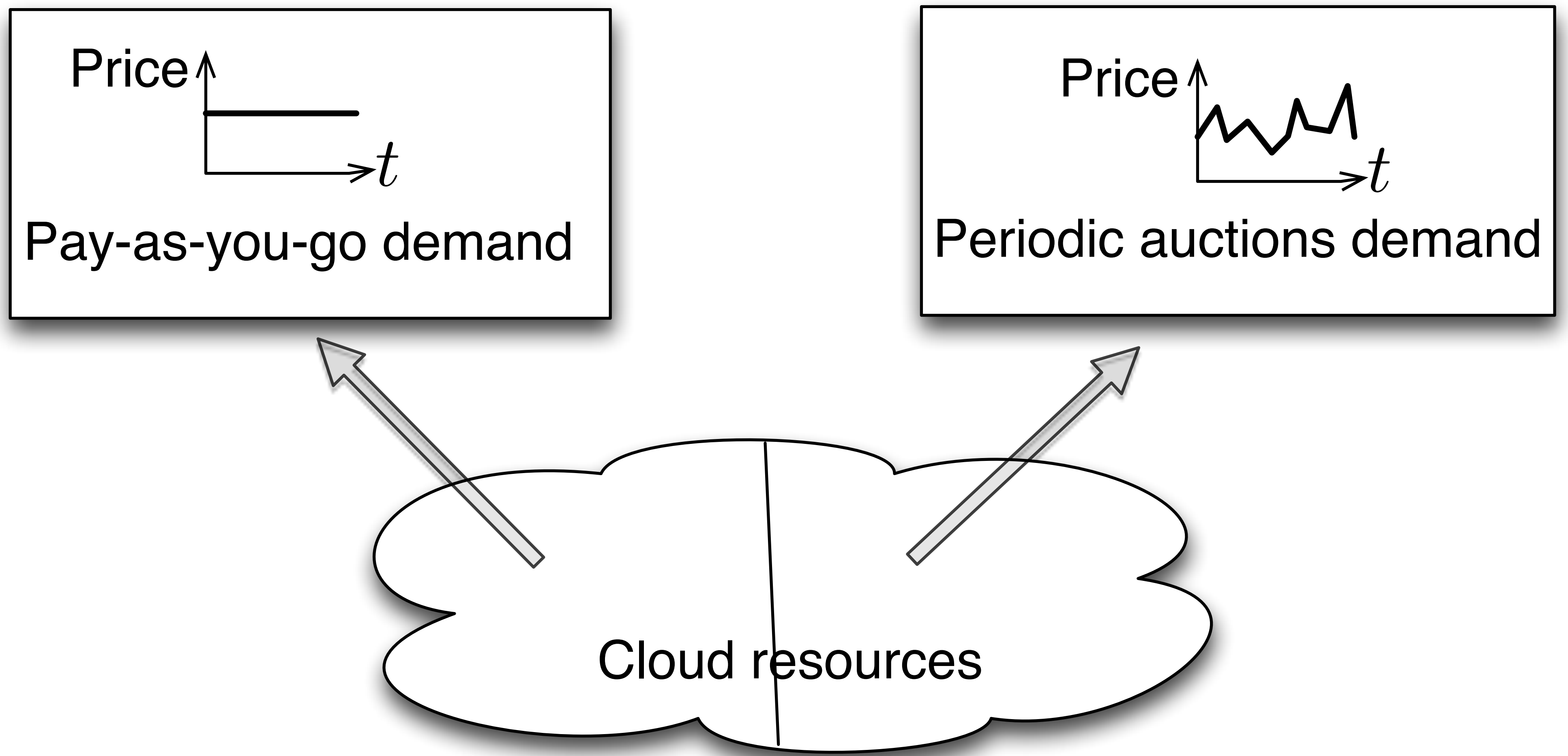
How many instances to offer in each channel in hour 1?



Strategy 1: Serve the on-demand user in hour 1 (revenue =$240)

Strategy 2: Strategically hold resources in hour 1 and serve the spot user in hour 2 (revenue = $300)

Our focus

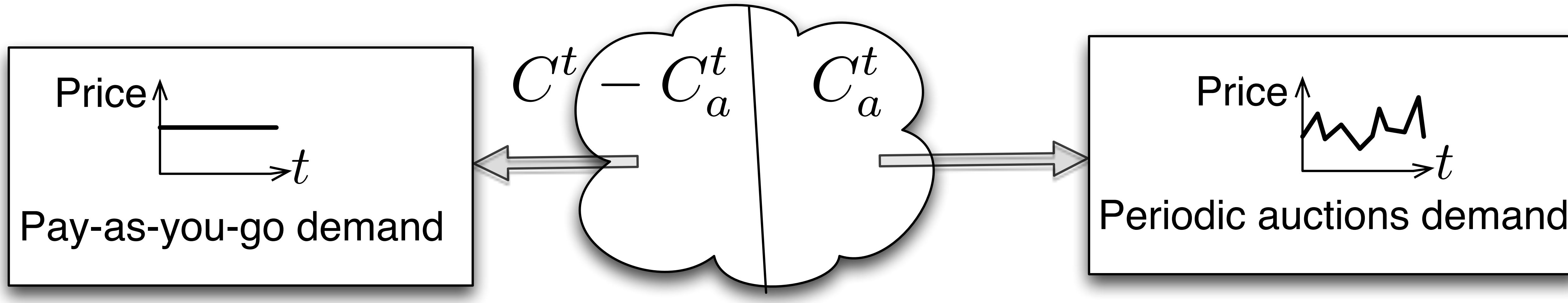


Dynamic capacity segmentation in two channels

On-demand channel with a fixed hourly rate

Periodic auction channel similar to EC2 spot market

Problem formulation



Cloud resources

$\Gamma^\tau(C^\tau)$: the optimal revenue collected during the prediction window

Auction revenue On-demand revenue

$$\Gamma^t(C^t) = \mathbf{E}\left[\max_{0\le \boxed{C_a^t} \le C^t} \left\{ \boxed{\gamma_a(C_a^t)} + \boxed{\gamma_r(C^t - C_a^t)} + \boxed{\mathbf{E}_{C^{t+1}}\left[\Gamma^{t+1}(C^{t+1})\right]} \right\}\right],$$

Future revenue

Revenue from the on-demand channel

q: the probability that a currently running on-demand instance is terminated by its user in the next time slot

Revenue from the on-demand channel

q: the probability that a currently running on-demand instance is terminated by its user in the next time slot

Revenue from the on-demand channel, with c instances allocated to it

$$\gamma_r(c) = \begin{cases} p_r c/q\,, & \text{if } c \leq R_r^t; \\ p_r R_r^t/q\,, & \text{otherwise,} \end{cases}$$

R_r^t: # of on-demand requests received at time t

A simple model yet gives interesting insights!

Periodic auctions

Auctions are carried out periodically

Each user *i* bids for computing instances

- True demand: n_i instances each with utility v_i
- Bid for r_i^t instances each at a price b_i^t
- (n_i, v_i) follows a joint p.d.f. $f_{n,v}$

A uniform clearing price p_a^t is posted in every time *t*

- User *i* wins if the bid exceeds the clearing price $b_i^t > p_a^t$

Upon losing, all running instances are terminated

Auction bidder

No partial fulfilment

- Lose all or win all
- The same as Amazon EC2 and other clouds

Utility function of bidder *i*

Gain Cost

$$u_i^t(r_i^t, b_i^t) = \begin{cases} \boxed{n_i v_i} - \boxed{r_i^t p_a^t}\,, & \text{if } p_a^t < b_i^t \text{ and } r_i^t \geq n_i; \\ 0\,, & \text{otherwise.} \end{cases}$$

What is the optimal auction mechanism?

Optimal auction design

(m+1)-price auction with a seller reservation price

Sort all bidders in a descending order of their bid prices, i.e., $b_1^t \geq b_2^t \geq \dots$

Reservation price = $\phi^{-1}(0)$, $\phi(v_i) = v_i - \dfrac{1 - F_v(v_i|n_i)}{f_v(v_i|n_i)}$

Optimal auction design

(m+1)-price auction with a seller reservation price

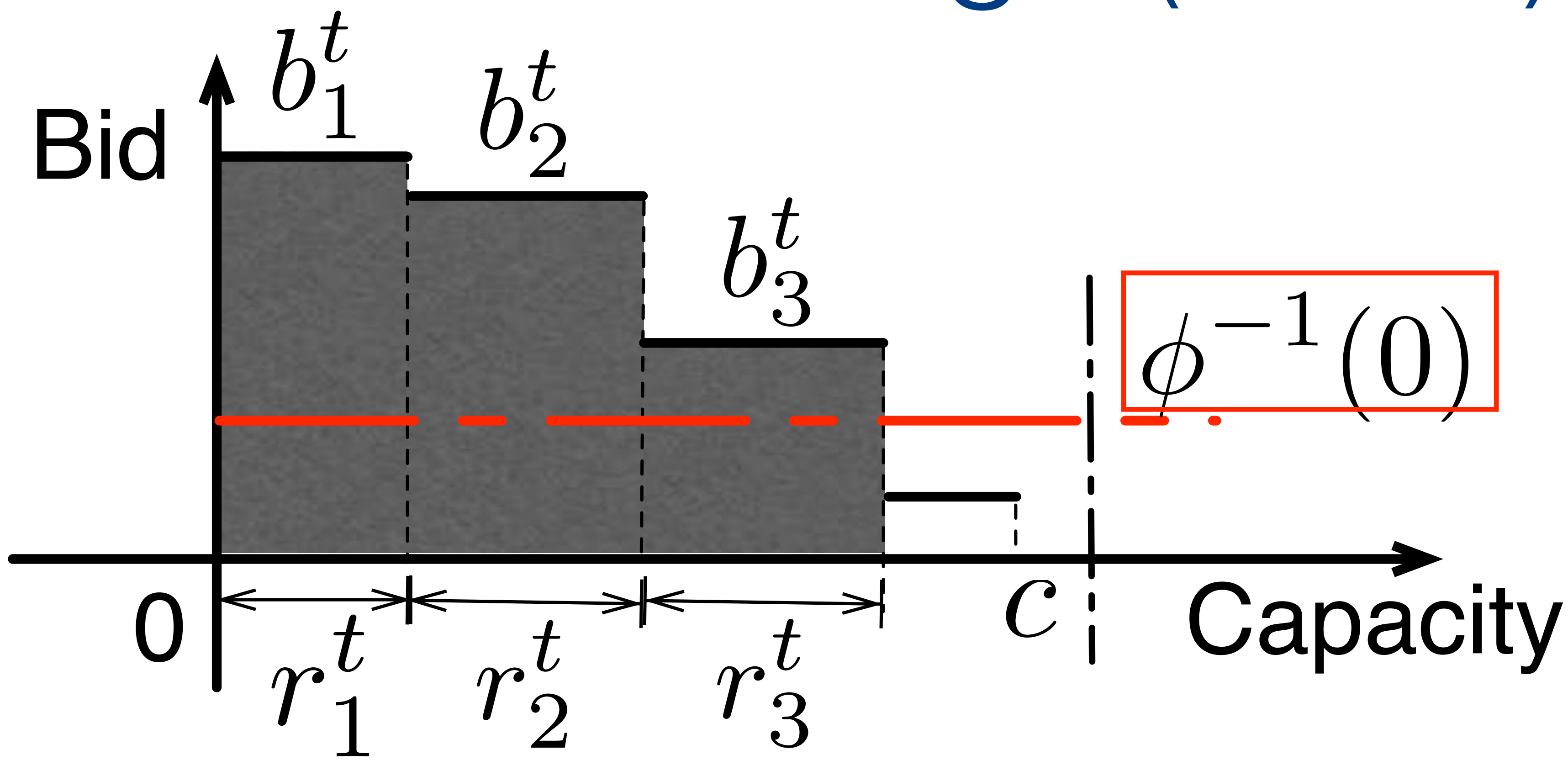
Sort all bidders in a descending order of their bid prices, i.e., $b_1^t \geq b_2^t \geq \ldots$

Reservation price = $\phi^{-1}(0)$, $\phi(v_i) = v_i - \dfrac{1 - F_v(v_i|n_i)}{f_v(v_i|n_i)}$

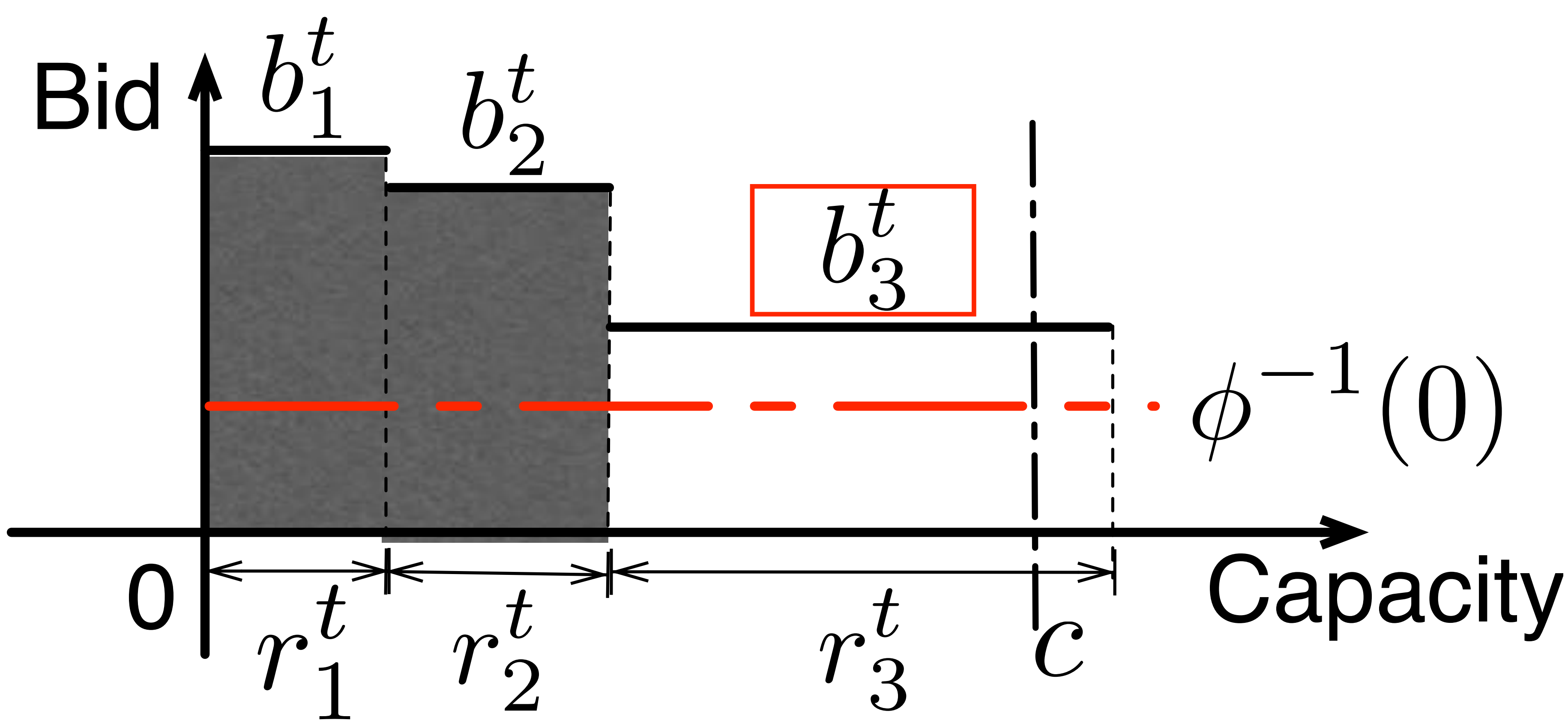
Keep accommodating top bidders, until (1) there is no available capacity to serve more or (2) no one bids higher than the reservation price. For the former case, winners are charged the highest bid of losers. For the later case, winners are charged the reservation price.

Optimal auction design (Cont.)

Case 1:



Case 2:



Optimal auction design (Cont.)

Proposition 1: The design maximizes the revenue among all auctions producing a uniform clearing price

Proposition 2: The design is two-dimensionally truthful

A user always reports true demand: $u_i^t(n_i, v_i) \geq u_i^t(r_i^t, b_i^t)$

Optimal auction design (Cont.)

Proposition 1: The design maximizes the revenue among all auctions producing a uniform clearing price

Proposition 2: The design is two-dimensionally truthful

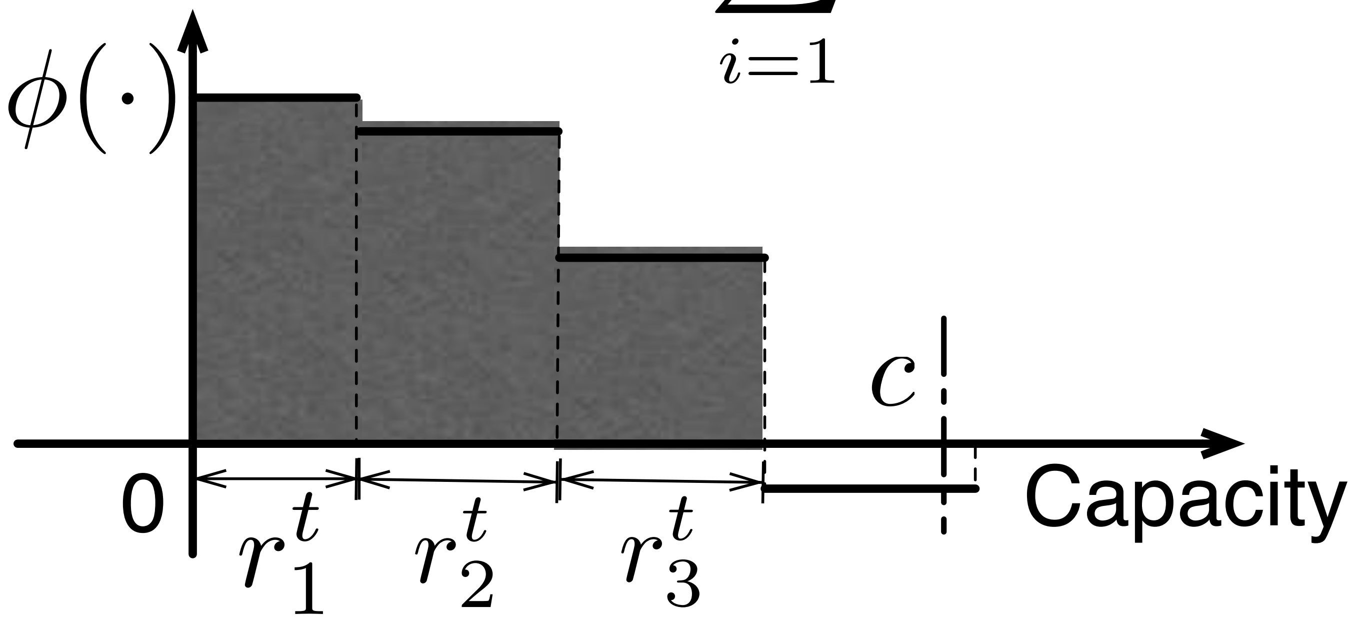
- A user always reports true demand: $u_i^t(n_i, v_i) \geq u_i^t(r_i^t, b_i^t)$

Remarks

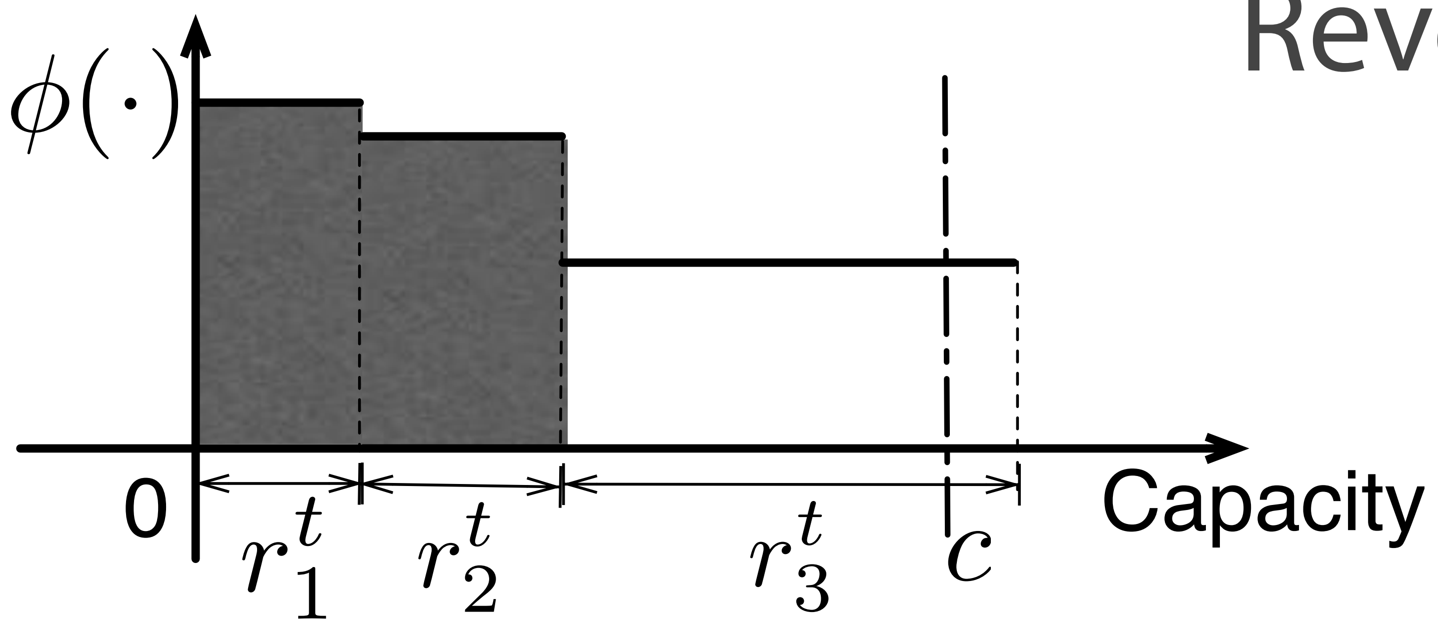
- Generally, (m+1)-price auction suffers from the problem of demand reduction and is neither truthful nor optimal when a bidder bids for multiple items
- We show that it is truthful and optimal in cloud markets where partial fulfilment is unaccepted

Auction revenue

Revenue: $\gamma_a(c) = \sum_{i=1}^{m} r_i^t \phi(b_i^t)$, where $\sum_{i=1}^{m} r_i^t \le c < \sum_{i=1}^{m+1} r_i^t$



Revenue = shaded area



Optimal capacity segmentation

Capacity segmentation revisit

Find the optimal segmentation C_a^t at time *t*

Auction On-demand

$$\Gamma^t(C^t) = \mathbf{E}\Bigg[\max_{0 \le C_a^t \le C^t} \Big\{\boxed{\gamma_a(C_a^t)} + \boxed{\gamma_r(C^t - C_a^t)} + \boxed{\mathbf{E}_{C^{t+1}}\left[\Gamma^{t+1}(C^{t+1})\right]}\Big\}\Bigg],$$

Future

State transition

$$C^{t+1} = C_a^t + X \quad X \sim B(C - C_a^t, k, q)$$

X: # of instances terminated by on-demand users at time *t*

Solving the capacity segmentation problem

$$\Gamma^t(C^t) = \mathbf{E}\left[\max_{0 \le C_a^t \le C^t} \left\{\gamma_a(C_a^t) + \gamma_r(C^t - C_a^t) + \mathbf{E}_{C^{t+1}}\left[\Gamma^{t+1}(C^{t+1})\right]\right\}\right],$$

$$C^{t+1} = C_a^t + X \quad X \sim B(C - C_a^t, k, q)$$

Direct solution is via numerical dynamic programming

- Hight computational complexity: $O(C^3)$
- C is the cloud capacity, and is usually huge
- Capacity segmentation is time sensitive: it has to be made in the beginning of every period

Approximation: solve the upper-bound problem

The upper-bound problem

$$\bar{\Gamma}^t(C^t) = \mathbf{E}\Bigg[\max_{0 \le C_a^t \le C^t} \Big\{\boxed{\bar{\gamma}_a(C_a^t)} + \gamma_r(C^t - C_a^t) + \mathbf{E}_X\left[\bar{\Gamma}^{t+1}(C_a^t + X)\right]\Big\}\Bigg].$$

$\bar{\gamma}_a(C_a^t)$: Revenue upper bound of the auction channel, calculated as if partial fulfilment is accepted in periodic auctions

The upper-bound problem

$$\bar{\Gamma}^t(C^t) = \mathbf{E}\left[\max_{0 \le C_a^t \le C^t} \left\{ \boxed{\bar{\gamma}_a(C_a^t)} + \gamma_r(C^t - C_a^t) + \mathbf{E}_X\left[\bar{\Gamma}^{t+1}(C_a^t + X)\right] \right\} \right].$$

Proposition 3: The upper-bound problem can be solved within $O(C^2)$

The upper-bound problem

$$\bar{\Gamma}^t(C^t) = \mathbf{E}\Bigg[\max_{0 \le C_a^t \le C^t} \Big\{\boxed{\bar{\gamma}_a(C_a^t)} + \gamma_r(C^t - C_a^t) + \mathbf{E}_X\left[\bar{\Gamma}^{t+1}(C_a^t + X)\right]\Big\}\Bigg].$$

Proposition 3: The upper-bound problem can be solved within $O(C^2)$

Intuition: previously calculated results can be reused in the following calculations

$\tilde{C}_a^\tau(C^\tau)$: optimal solution to the upper-bound problem

$$\tilde{C}_a^\tau(C^\tau + 1) - 1 \le \tilde{C}_a^\tau(C^\tau) \le \tilde{C}_a^\tau(C^\tau + 1).$$

The approximation

We solve the upper-bound problem and offer $\tilde{C}_a^t(C^t)$ instances in the auction channel at time *t*

$\tilde{\Gamma}^t$: revenue of the approximate solution

The approximation

We solve the upper-bound problem and offer $\tilde{C}_a^t(C^t)$ instances in the auction channel at time *t*

$\tilde{\Gamma}^t$: revenue of the approximate solution

Proposition 4 (asymptotic optimality): $\tilde{\Gamma}^t \to \Gamma^t$ w.p. 1 if the number of bidders $N_a^\tau \to \infty$ for all $\tau = t, \ldots, t + w$

The approximation

We solve the upper-bound problem and offer $\tilde{C}_a^t(C^t)$ instances in the auction channel at time *t*

$\tilde{\Gamma}^t$: revenue of the approximate solution

Proposition 4 (asymptotic optimality): $\tilde{\Gamma}^t \to \Gamma^t$ w.p. 1 if the number of bidders $N_a^\tau \to \infty$ for all $\tau = t, \ldots, t+w$

Remarks

The condition $N_a^\tau \to \infty$ is naturally satisfied in cloud environments as there are always a large amount of cloud users requesting computing instances

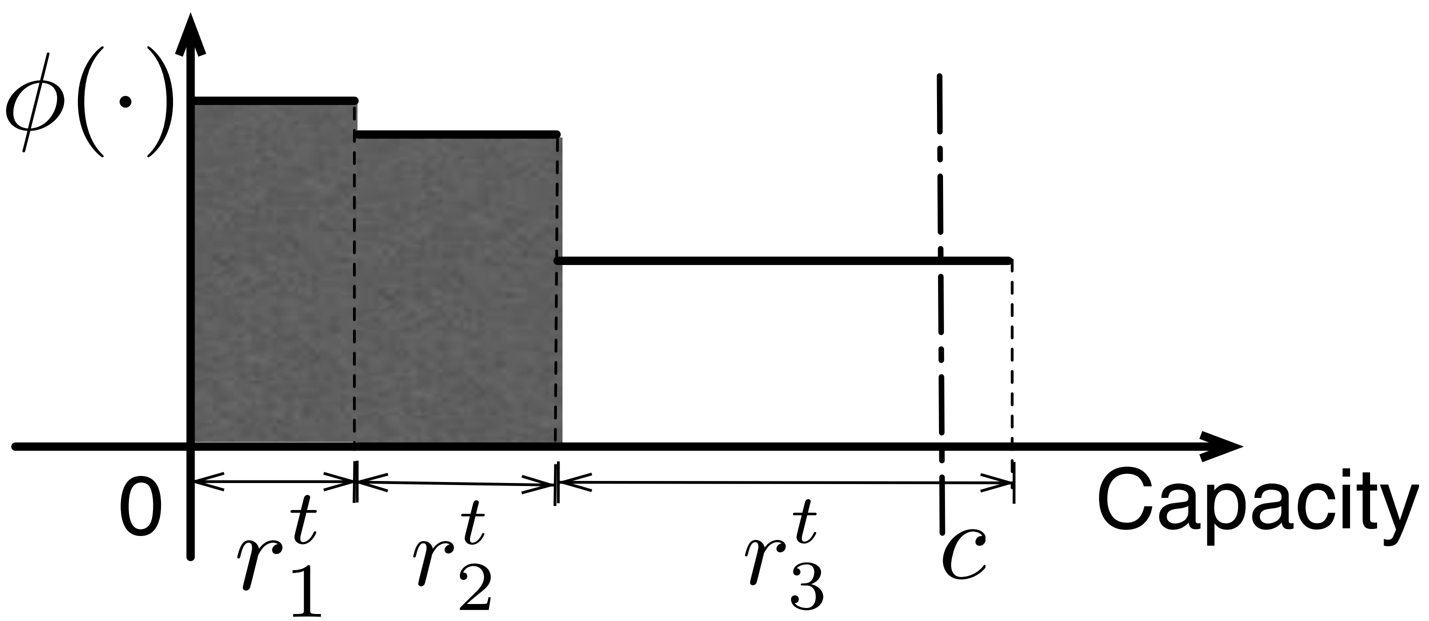
Asymptotically optimal solution

We turn to an efficient approximate solution

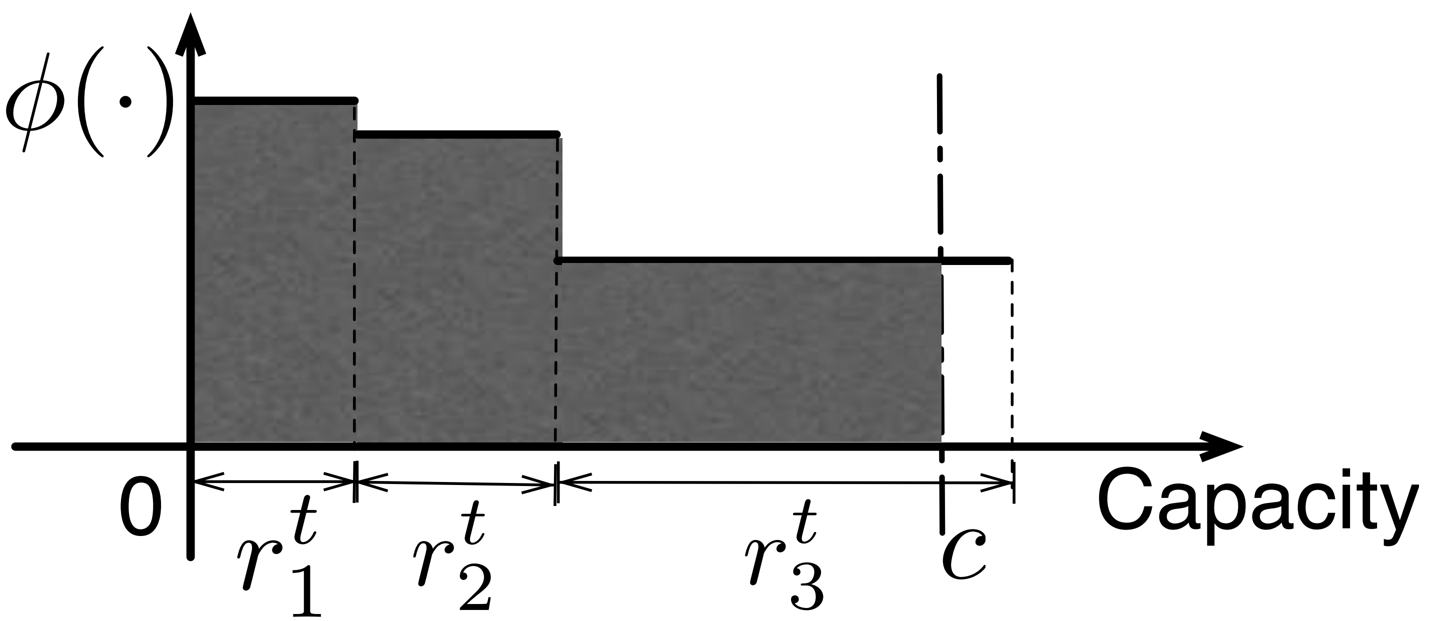
- Proved to be asymptotically optimal
- Almost optimal in simulations: performance gap < 2%
- Highly efficient, with time complexity $O(C^2)$

Auction revenue upper bound

$\gamma_a(c)$



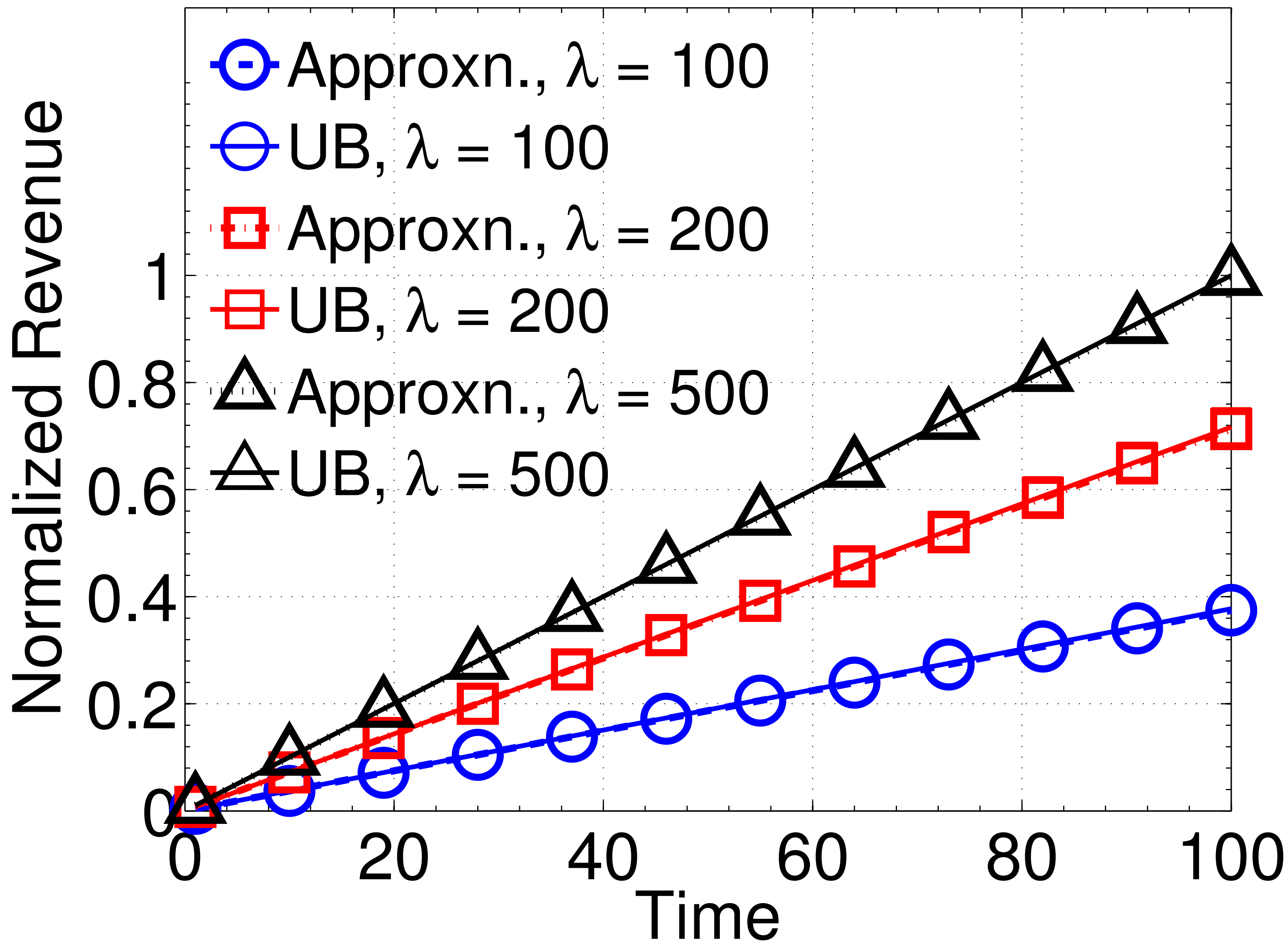
$\bar{\gamma}_a(c)$



As if partial fulfilment is accepted

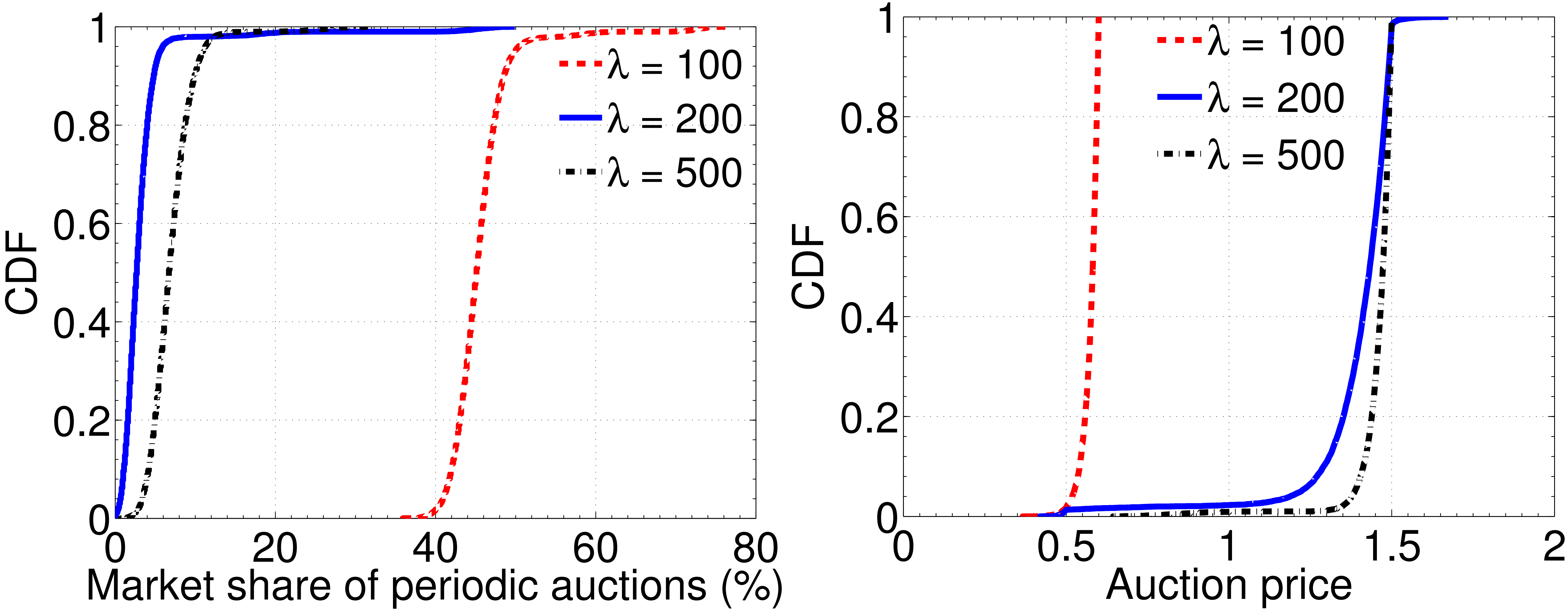
Evaluations

Revenue performance



Users arrive into the two pricing channels following a Poisson process, with intensity being low ($\lambda = 100$), medium ($\lambda = 200$), and high ($\lambda = 500$).

Market share and the clearing price



Conclusions

We investigate the optimal capacity segmentation problem with hybrid cloud pricing.

We show that optimal periodic auctions are of the form of (m+1)-price auction with a seller reservation price.

We design an efficient capacity segmentation scheme that is proved to be asymptotically optimal.

Simulation studies show that the solution is almost optimal.

Thank you!

http://iqua.ece.toronto.edu/